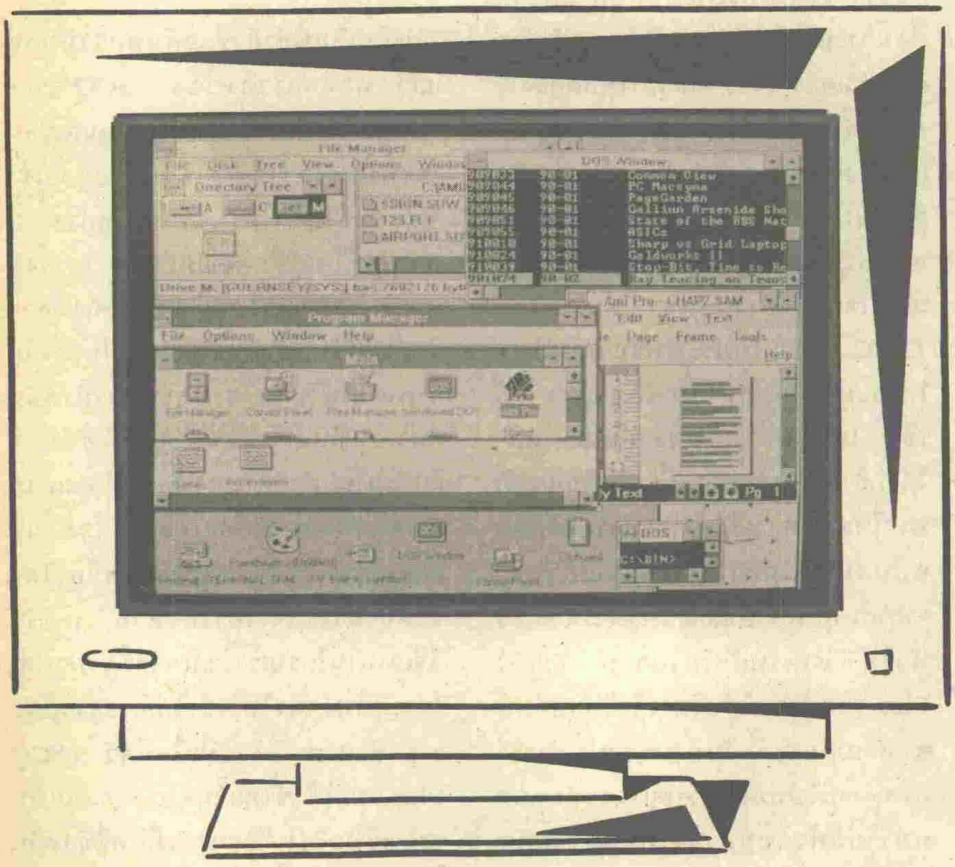
ยูสเซอร์อินเตอร์เฟส -ระบบวินโดว์

พัฒนาการวิธีเชื่อมต่อผู้ใช้กับโปรแกรม

ความพยายามที่จะหาวิธีการเชื่อมโยงระหว่างคนกับคอมพิวเตอร์ในรูปกราฟิกที่ใช้ได้ง่ายและก่อให้เกิดระบบวินโดว์ที่เรียกว่า GUI



เริ่มจากความต้องการของผู้ใช้คอมพิวเตอร์ ที่ต้องการรูปแบบที่เป็นธรรมชาติง่ายต่อการเข้าใจ จึงได้มีการพัฒนาวิธีการสั่งงานคอมพิวเตอร์ในรูปแบบต่าง ๆ กัน เริ่มตั้งแต่การเขียนเมนูให้เลือกตามรายการ การใช้เมนูในยุคแรกนี้ยังไม่ประทับใจเพราะบนจอภาพยังแสดงได้แต่เพียงตัวอักษรแต่เพียงอย่างเดียว ประกอบกับความรู้สึกของผู้ใช้อยากจะเห็นข้อความหรืองานพร้อมกัน ดังนั้นจึงมีความพยายามที่จะหาวิธีการเชื่อมโยงระหว่างคนกับคอมพิวเตอร์ในรูปกราฟิกที่ใช้ได้ง่าย

จุดเริ่มต้นของโครงการยูสเซอร์อินเตอร์เฟส -ระบบวินโดว์

จุดเริ่มต้นเกิดขึ้นจากกลุ่มนักวิจัยของบริษัทเซล็อก ที่ศูนย์วิจัยพาโลอัลโต เริ่มโครงการที่ชื่อว่า Xerox star มีการสร้างระบบยูสเซอร์อินเตอร์เฟสบนเครื่องคอมพิวเตอร์ของเซล็อก เพื่อใช้ในงานทางด้านระบบการเรียงพิมพ์และงานจำพวกเพจเมกเกอร์ ต่อมาสตีฟจ็อบได้มีโอกาสไปเยี่ยมเยียนที่ศูนย์วิจัยนี้ และพบกับผู้วิจัยโครงการนี้โดยตรง สตีฟจ็อบชอบใจในวิธีการใช้งานของเครื่องเซล็อกมาก จึงหันกลับมาเริ่มโครงการพัฒนาไมโครคอมพิวเตอร์แมคอินทอชตามลักษณะยูสเซอร์อินเตอร์-

เฟสแบบเดียวกัน แมคอินทอชประสบผลสำเร็จได้ดีพอควร และเป็นระบบไมโครคอมพิวเตอร์ระบบเดียวกันที่มาแข่งขันกับไมโครคอมพิวเตอร์ในตระกูลพีซีได้ จุดเริ่มในปี 2527 ของแมคอินทอชจึงถือได้ว่าระบบวินโดว์ได้เริ่มต้นแพร่หลายอย่างจริงจังแล้ว

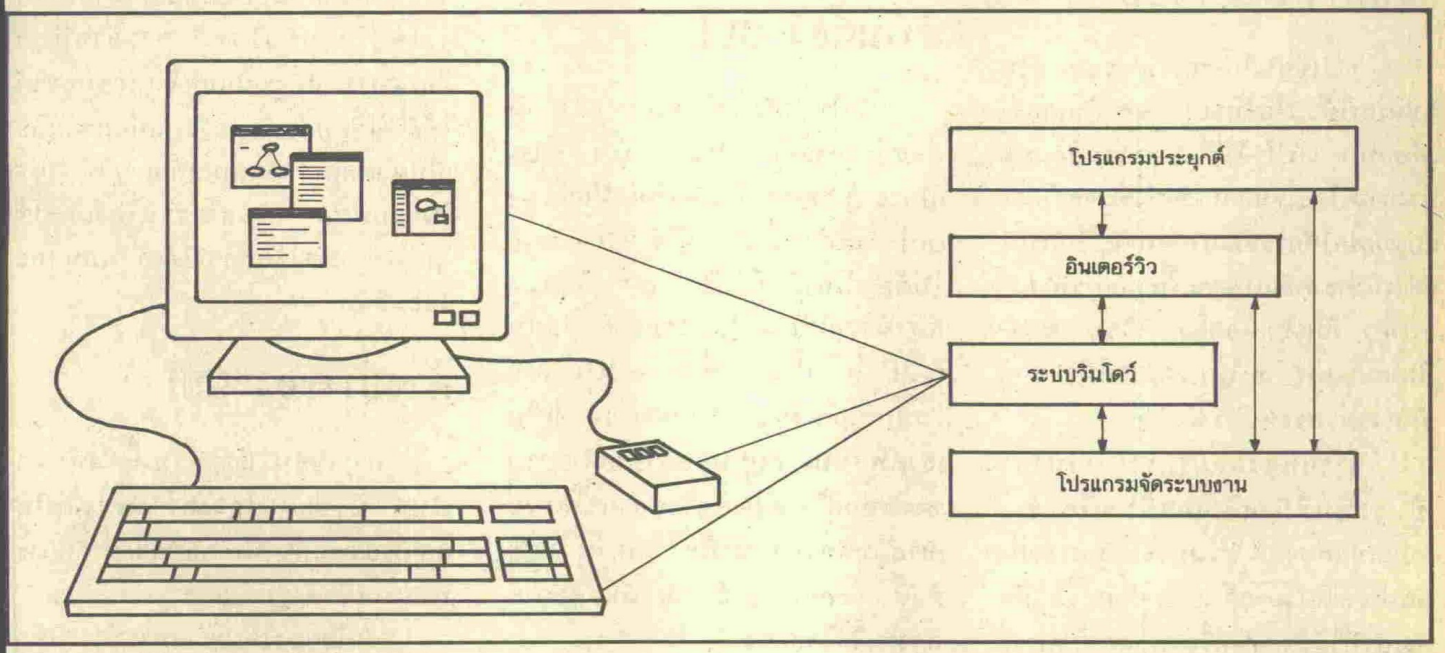
เวลาเป็นเครื่องพิสูจน์

หลังจากที่แมคอินทอชได้แนะนำระบบวินโดว์มาถึงห้าปี ไอบีเอ็มก็เสนอโครงสร้างซอฟต์แวร์ที่ติดมากับโอเอสทูที่เรียกว่า PM-Presentation Manager ซึ่งก็ไม่เหมือนกับแมคทีเดียว แต่ก็จัดว่าเป็นระบบยูสเซอร์อินเตอร์เฟสในระบบวินโดว์ ไอบีเอ็มยังเสนอวิธีการเชื่อมต่อเพื่อกำหนดรูปแบบที่เป็นมาตรฐานขึ้น เพื่อให้ผู้พัฒนาโปรแกรมเรียกเข้าหาระบบได้ดียิ่งขึ้น และเรียกระบบนี้ว่า SAA-System Application Architecture (SAA) ไมโครซอฟต์ก็ได้พัฒนาระบบไมโครซอฟต์วินโดว์ของตนเองให้ดียิ่งขึ้น ในขณะเดียวกับมหาวิทยาลัย MIT ก็พัฒนาวินโดว์ระบบเอกซ์ นอกจากนี้ก็มีโครงการในลักษณะนี้เกิดขึ้นมามากมาย เช่น ในญี่ปุ่นก็ผนวกเป็นโครงการอยู่ในโครงการ TRON และตั้งชื่อว่า BTRON สตีฟจ๊อบกลับออกมาตั้งบริษัทใหม่พัฒนาเครื่อง NEXT และพัฒนาระบบยูสเซอร์อินเตอร์เฟสชื่อ NextStep

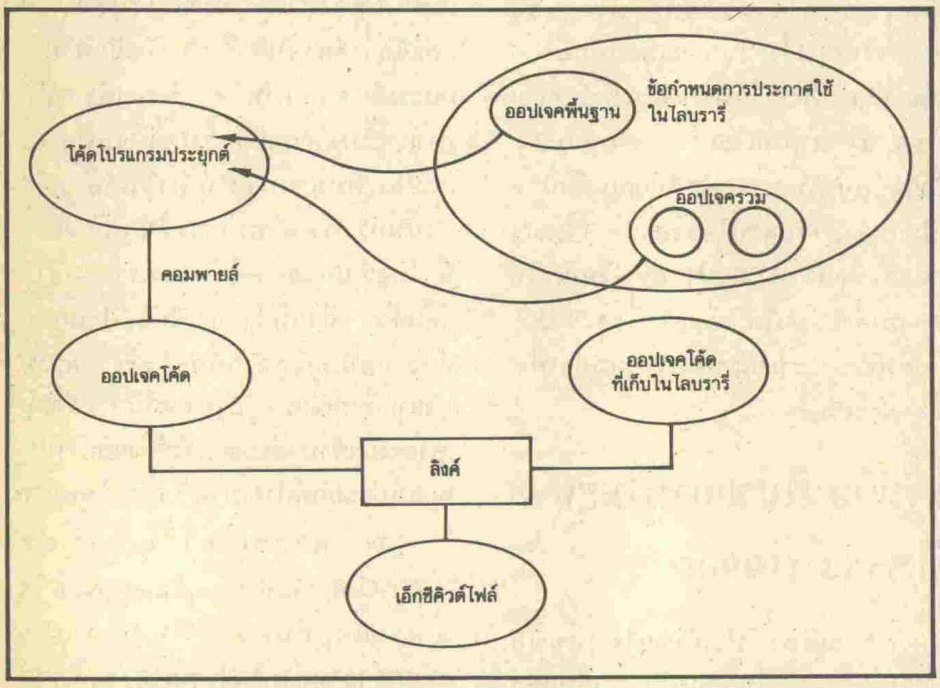
การเขียนโปรแกรมในยุคปี 2533 (1990)

การเขียนโปรแกรมในยุคสมัยปัจจุบันเริ่มแตกต่างจากอดีตมากขึ้นแล้ว โปรแกรมประยุกต์ในปัจจุบันต้องการกราฟิกยูสเซอร์อินเตอร์เฟสและหลักการของออปเจคโอเรียนเต็ดโปรแกรมมิ่ง การเขียนโปรแกรมในปัจจุบันต้องการเทคนิคเพื่อให้ได้งานที่สลับซับซ้อนขึ้น และจัดว่าต้องอาศัยศิลปะการเขียนโปรแกรมอีกรูปแบบหนึ่ง ภาษาซีจะเป็นภาษาที่มีบทบาทต่อการพัฒนาโปรแกรมและจะเพิ่มบทบาทมากยิ่งขึ้นต่อ ๆ ไป ดังนั้นการเขียนโปรแกรมในยุคปี 2533 นี้ คงเริ่มด้วยการใช้หลักการดังกล่าวได้แล้ว ถึงแม้ว่าการเขียนโปรแกรมด้วยเทคนิคดังกล่าวนี้จะยังสร้างความยุ่งยากต่อผู้เขียนอยู่บ้าง แต่ก็มีการพัฒนาเครื่องมือเข้ามาช่วยจะทำให้ผู้พัฒนาโปรแกรมประยุกต์ทำงานได้เร็วและดีขึ้นมาก

กราฟิกยูสเซอร์อินเตอร์เฟส (GUI-Graphical User Interface) จึงเป็นรูปแบบที่มีหน่วยงานกลางหลายแห่งพยายามกำหนดเป็นมาตรฐาน เช่น IEEE, ISO อย่างไรก็ตาม แนวทางของ GUI ที่พัฒนาในเครื่องแมคอินทอช ไมโครซอฟต์เอกซ์วินโดว์ หรือโอเอสทูพีเอ็ม ก็มีแนวทางที่คล้ายกัน ดังนั้นโปรแกรมประยุกต์จึงเหมาะกับผู้ใช้ เพราะผู้ใช้จะคุ้นเคยไปในทางเดียวกัน



รูปที่ 1 ระดับขั้นของซอฟต์แวร์ประยุกต์



รูปที่ 2 การเขียนโปรแกรมประยุกต์เชื่อมต่อกับระบบวินโดว์

โครงสร้างของยูสเซอร์อินเตอร์เฟสระบบวินโดว์ (GUI)

การเขียนโปรแกรมประยุกต์ในรูปแบบใช้วินโดว์และยูสเซอร์อินเตอร์เฟสแบบ GUI ได้มีแนวความคิดและพัฒนามาโดยใช้หลักการแบ่งชั้นของโปรแกรมเพื่อให้การพัฒนาซอฟต์แวร์เป็นไปอย่างมีประสิทธิภาพและง่ายต่อการพัฒนา รูปที่ 1 เป็นโมเดลของการใช้ยูสเซอร์อินเตอร์เฟส โดยให้โปรแกรมประยุกต์จัดการผ่านระดับชั้นต่าง ๆ

การแสดงผลผ่านระบบวินโดว์เป็นระบบที่ผ่านอินเตอร์วิวเพื่อสร้างรูปแบบแสดงผล ส่วนนี้ใช้หลักการของออปเจคโอเรียนเต็ดผ่านทางเครื่องมือพัฒนาที่ได้จัดการเครื่องมือและไลบรารี่ต่าง ๆ ไว้ให้แล้ว การเชื่อมโยงระหว่างโปรแกรมประยุกต์กับออปเจคต่าง ๆ ของอินเตอร์วิวในระบบวินโดว์ แสดงดังรูปที่ 2

ทำไมต้อง GUI

หัวใจสำคัญของระบบ GUI มีสองประการคือ ประการแรก GUI เป็นระบบยูสเซอร์อินเตอร์เฟสที่อยู่นอกเหนือจากส่วนอื่นใด ผู้ใช้ GUI อาจไม่ต้องเป็นนักเขียนโปรแกรมมืออาชีพก็สามารถเขียนโปรแกรมเชื่อมต่อกับ GUI ได้ ประการที่สอง GUI เป็นระบบที่แสดงความรู้สึกตามที่มองเห็นเอาต์พุตที่ปรากฏบนจอภาพบอกความหมายต่อผู้ใช้ได้ง่าย โดยสามารถรวมเอาส่วนของตัวหนังสือ รูปภาพ และตัวชี้ (icon) มาสัมพันธ์กันและผู้ใช้สั่งงานได้โดยตรง

ปกติผู้เขียนโปรแกรมประยุกต์มักมีจุดอ่อนในเรื่องการเขียนให้มีระบบยูสเซอร์อินเตอร์เฟสที่ดี ผู้เขียนโปรแกรมจะเน้นและเข้าใจในระบบประยุกต์ได้ดี แต่ขาดเทคนิคหรือวิธีการในการเชื่อมต่อทางยูสเซอร์อินเตอร์เฟส ดังนั้นการใช้ GUI จึงเป็นทางออก เพราะ GUI ได้รับการพัฒนามาเพื่อผู้ใช้โดยตรง การเชื่อมต่อกับ GUI จึงทำให้โปรแกรมประยุกต์นั้นน่าใช้ขึ้นอีกมาก

ก่อนอื่นต้องทำความเข้าใจว่า GUI ไม่ใช่โปรแกรมจัดระบบงาน แต่เป็นโปรแกรมช่วยจัดการสภาวะแวดล้อมต่าง ๆ ให้ GUI จะเป็นตัวเชื่อมกับระบบโอเอส แต่อย่างไรก็ตาม GUI บางโปรแกรมเช่นในแมคอินทอช ถูกเชื่อมกับโอเอสอย่างแนบแน่นยากที่จะแบ่งแยกออกมาต่างหากจากโปรแกรมได้ GUI บางตัวก็เสมือนโปรแกรมแพคเกจแยกออกมาต่างหากจากโปรแกรมจัดระบบงานอย่างชัดเจน จนคล้ายเป็นโปรแกรมแยกจากกันดังเช่นโปรแกรมไมโครซอฟต์วินโดว์

โปรแกรม GUI ตัวที่มีผลทำให้ยูสเซอร์ยอมรับในหลักการมากที่สุดคือ GUI ที่อยู่บนแมคอินทอชและได้ให้ผู้ใช้เชื่อมต่อผ่านทางทูลบ๊อกซ์ และเป็นผลต่อการสร้างแนวคิดการใช้ GUI ในระบบอื่น การสร้างวินโดว์และใช้วินโดว์ การใช้อุปกรณ์ตัวชี้ เมนูและไดอะล็อก

ตัวอย่างของ GUI

การเขียนโปรแกรมเชื่อมกับตัว GUI มีรูปแบบโครงสร้างพื้นฐานในลักษณะเดียวกัน ประกอบด้วยลำดับการทำงานหลักดังรูปที่ 3

หัวใจของโครงข่ายการทำงานคือ ลูปกับทำงานของ event โปรแกรมจะ

ต้นฉบับไม่มีหน้า 237-238

No This Page 237-238 In Original

ยูสเซอร์อินเตอร์เฟส-ระบบวินโดว์
พัฒนาการวิธีเชื่อมต่อผู้ใช้กับโปรแกรม

```
program and GUI initialization
  loop
      wait for event
      process event
  until guit event received
program and GUI clean up
```

รูปที่ 3 การเขียนโปรแกรมเชื่อมต่อกับ GUI

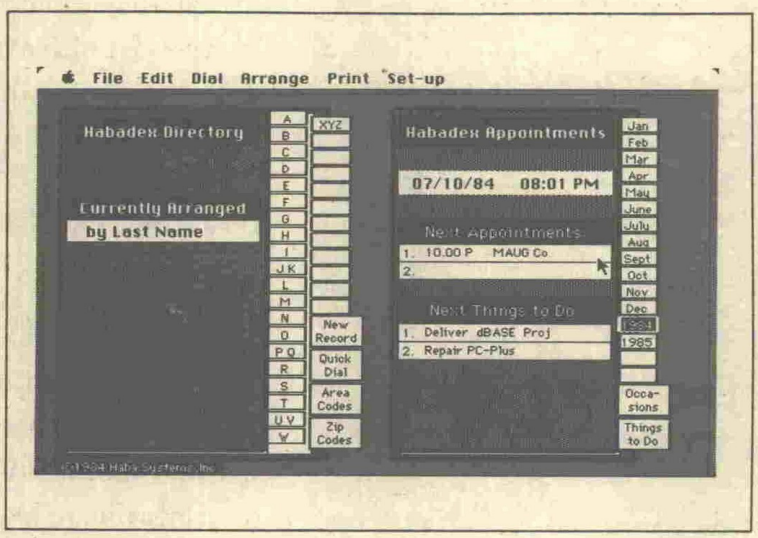
รอตรวจสอบสำหรับ event ที่เกิดขึ้น และประมวลผลตาม event event จะได้รับการสร้างขึ้นผ่าน GUI เช่น เมื่อผู้ใช้คลิกเมาส์เพื่อเลือกเมนูหรือเลือกกดปุ่มที่จะติดต่อกับระบบการทำงานจะตอบสนองตาม event นั้น ๆ

สิ่งที่เป็นความสำเร็จของระบบวินโดว์ในระบบ GUI คือโปรแกรมสามารถเปิดได้หลายวินโดว์ โดยทำให้การทำงานในระบบหลายโปรแกรมในหลายวินโดว์ โดยที่ผู้ใช้สามารถกำหนดให้แอคตีฟที่วินโดว์ใดก็ได้ดังรูปที่ 4

GUI ในแมคอินทอช

ระบบยูสเซอร์อินเตอร์เฟสของแมคอินทอช เป็นส่วนหนึ่งของโปรแกรมที่อยู่ในแมคอินทอชทูลบ๊อกซ์ ทูลบ๊อกซ์ของแมคอินทอช เป็นโปรแกรมเก็บไว้ในไลบรารี่ที่อยู่ใน ROM ขนาดของทูลบ๊อกซ์นี้มีขนาด 64 กิโลไบต์ อย่างไรก็ตามได้มีการเพิ่มเติมเวอร์ชันของ ROM ออกไปมากแล้ว แต่ได้สร้างระบบพื้นฐานการเรียกใช้ยังคงเหมือนเดิม โปรแกรมที่เขียนขึ้นเพื่อใช้กับระบบแมคอินทอชตั้งแต่เวอร์ชันแรกในปี พ.ศ.2527 ก็ยังใช้ได้กับเครื่องแมคอินทอชในปัจจุบัน บริษัทแอปเปิ้ลต้องทำงานอย่างหนัก เพื่อให้ระบบซอฟต์แวร์ที่ใช้เป็นแบบคอมแพตทิเบิ้ลกันมาตลอดตั้งแต่เริ่มผลิตเครื่องแมคอินทอช

ข้อเด่นของระบบ GUI ของแมคอินทอช ที่เห็นเด่นชัดที่นำเอาโปรแกรมมาไว้ใน ROM ทำให้ผู้พัฒนาโปรแกรมหรือโปรแกรมประยุกต์ที่เก็บอยู่ใน RAM มีขนาดเล็กลง การเขียนโปรแกรมบนแมคอินทอชจึงต้องอาศัยทูลบ๊อกซ์ GUI นี้ รูปแบบของซอฟต์แวร์ประยุกต์จึงเป็นแบบเดียวกันหมด ทำให้การใช้งานไม่สับสน



รูปที่ 4 ระบบวินโดว์ในแมคอินทอช ทำให้การพัฒนาซอฟต์แวร์ประยุกต์ง่ายขึ้นและขนาดของโปรแกรมประยุกต์เล็กลงเพราะสามารถเรียก GUI ที่อยู่ใน ROM ได้

การใช้ ROM มีข้อเสียตรงส่วนที่การปรับปรุงเวอร์ชันทำได้ยาก การแก้ไขโปรแกรมใน ROM ในกรณีที่พบบักก็ทำได้ยาก เพราะต้องเปลี่ยนแปลง ROM ใหม่ โปรแกรม GUI ของแมคอินทอชก็เป็นโปรแกรมที่ค่อนข้างจะมีประสิทธิภาพ มีการจัดสรรทรัพยากรให้ใช้งานอย่างดี

แมคอินทอชจัดการกับ event ในระบบโกลบอล ซึ่งเป็นข้อแตกต่างจากระบบวินโดว์ของระบบอื่น ซึ่งจัดการ event ภายในแต่ละวินโดว์ เช่น ระบบของเอมิกา หรือจัดแบ่งเป็นคลาสในกรณีของไมโครซอฟต์วินโดว์ ทั้งนี้เพราะแมคอินทอชออกแบบระบบวินโดว์ของตนในลักษณะงานแบบซิงเกิ้ลทาสก์ แต่ระบบไมโครซอฟต์วินโดว์ออกแบบมาในลักษณะงานแบบมัลติทาสก์

ไมโครซอฟต์วินโดว์

บนพีซีซึ่งแต่เดิมไอบีเอ็มไม่ได้ออกแบบมาสำหรับใช้งานในลักษณะ GUI แต่เมื่อระบบกราฟิกดีขึ้นมาก ไมโครซอฟต์ก็ได้พัฒนาโปรแกรมไมโครซอฟต์วินโดว์ขึ้น ระบบไมโครซอฟต์วินโดว์จึงเป็นโปรแกรมที่ทำงานอยู่เหนือดอสอีกขั้นหนึ่ง ซึ่งเป็นขั้นที่โปรแกรมประยุกต์เดิมใช้งานอยู่แล้ว และเพื่อให้ผู้ออกแบบโปรแกรมประยุกต์ทำงานได้ภายใต้วินโดว์ ไมโครซอฟต์จึงพัฒนาซอฟต์แวร์ที่เป็นเครื่องมือสำหรับงานนี้โดยตรง คือ SDK-Software Development Kit ซึ่งจะต้องใช้โปรแกรมย่อยหรือไลบรารี่จากชุดคิทนี้

ไมโครซอฟต์วินโดว์เป็นระบบ GUI ที่ค่อนข้างจะซับซ้อนมากกว่าของแมคอินทอช แต่ไม่ได้หมายความว่า

ดีกว่า เพียงแต่มีออปชันให้เลือกใช้งานได้มากขึ้น ผู้เขียนโปรแกรมบนไมโครซอฟต์วินโดว์จะต้องรู้เรื่องต่าง ๆ ได้ดีและรู้จักกับการเรียกใช้ SDK สิ่งที่สำคัญของการเขียนโปรแกรมในไมโครซอฟต์วินโดว์คือ การใช้หลักการของออปเจกโอเรียนเต็ดโปรแกรมมิ่งมีการแบ่งคลาสของวินโดว์ มีการส่งย้ายโปรโตคอลระหว่างออปเจค

เช่นเดียวกับระบบของแมคอินทอช คือ วินโดว์ของไมโครซอฟต์สนับสนุนอุปกรณ์อินพุตเอาต์พุตที่อิสระมากขึ้น สามารถใช้กับอุปกรณ์เครื่องพิมพ์ ใช้กับจอภาพแบบต่าง ๆ ได้ การใช้งานในระบบวินโดว์นั้นควรจะใช้กับเมาส์ แต่ถ้าหากไม่มีก็สามารถใช้กับคีย์บอร์ดได้ ดังรูปที่ 5

ไอบีเอ็มก็พัฒนา PM

PM-Presentation Manager เป็นระบบ GUI ที่ได้รับการพัฒนามาใช้กับโอเอสทู ระบบไมโครซอฟต์วินโดว์กับพีเอ็ม มีลักษณะที่คล้ายคลึงกันมากในความรู้สึกของผู้ใช้ แต่ในระดับซอร์สโค้ดนั้นไม่สามารถนำมาใช้ร่วมกันได้

พีเอ็มของโอเอสทูมีขีดความสามารถที่พัฒนามาให้เหมาะสมกับโอเอสในระบบมัลติทาสก์ของโอเอสทูโดยตรง การเขียนโปรแกรมก็เป็นโครงสร้างคล้าย ๆ กับระบบไมโครซอฟต์วินโดว์ ดังนั้นการเขียนโปรแกรมในไมโครซอฟต์วินโดว์แล้วย้ายมาพัฒนาบนพีเอ็มจะง่ายกว่าการพัฒนาบนแมคอินทอชแล้วย้ายมาที่พีเอ็ม

ไดนามิกส์ลิงค์ไลบรารี่:

สิ่งที่เพิ่มขีดความสามารถในการใช้หน่วยความจำจำกัด

หากพิจารณาระบบวินโดว์ในระบบ GUI ในลักษณะมัลติทาสก์ของไมโครซอฟต์วินโดว์หรือของพีเอ็ม เป็นเรื่องที่เป็นปัญหาต่อการใช้หน่วยความจำอย่างยิ่ง เช่นถ้าผู้ใช้สร้างวินโดว์ขึ้นมาสามวินโดว์ แต่ละวินโดว์เรียกโปรแกรมประยุกต์ขึ้นมา ซึ่งอาจจะเป็นโปรแกรมเดียวกันหรือต่างโปรแกรมก็ได้ ปัญหาจะมีอะไรบ้าง

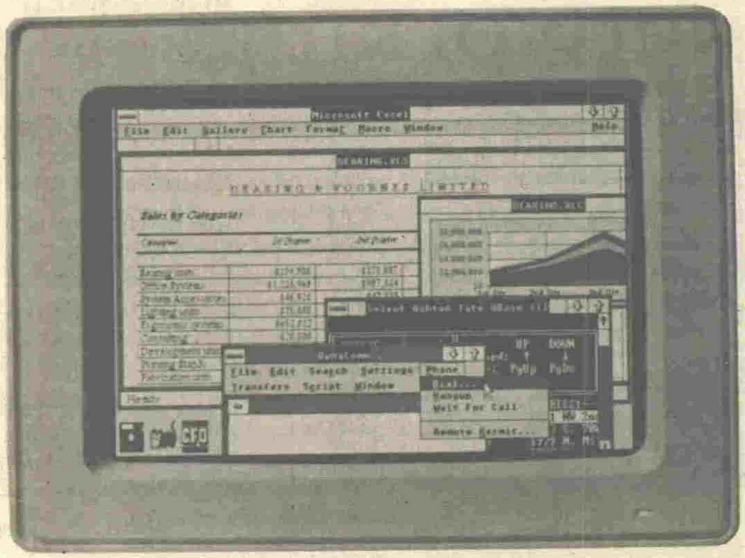
ปัญหาข้อแรกคือ ถ้าหากให้ทั้งสามโปรแกรมประยุกต์ได้รับการโหลดมายังหน่วยความจำ การใช้หน่วยความจำคงต้องมีขนาดใหญ่มาก ซึ่งก็เป็นไปได้ยากสำหรับการใช้กับฮาร์ดแวร์ปัจจุบัน ดังนั้นจึงต้องใช้ระบบสแว็ป (SWAP) คือโหลดโปรแกรมไปมาเมื่อแอคตีฟวินโดว์ต่างกัน ซึ่งก็ต้องเสียเวลา

ปัญหาข้อต่อมาก็คือ ในแต่ละโปรแกรมประยุกต์อาจมีการใช้โมดูลในไลบรารี่ร่วมกัน การโหลดไปมาหรือการรวมไลบรารี่โมดูลที่ใช้ไว้ในโปรแกรมประยุกต์ต่าง ๆ ก็จะทำให้ขนาดของโปรแกรมทั้งหมดใหญ่ เพราะโมดูลในไลบรารี่ได้รับการลิงค์ไปเก็บไว้ในโปรแกรมประยุกต์ทั้งหมด

ดังนั้นการแก้ปัญหาของการใช้ Run time library ที่ต้องลิงค์เข้าไปในโปรแกรมประยุกต์ต้องเปลี่ยนไป โดยให้เป็นการลิงค์ในรูปแบบที่เรียกว่าไดนามิกส์ลิงค์ไลบรารี่ กล่าวคือ เมื่อใช้ในระบบวินโดว์ หลายโปรแกรมสามารถแบ่งการใช้บางโมดูลร่วมกันขณะกำลังทำงานได้ การลิงค์จะไม่นำเอาโมดูลในไลบรารี่เข้าไปรวมกับโมดูลของโปรแกรมประยุกต์ ดังรูปที่ 6 และ 7

การพัฒนาต้องอาศัยทูลคิท

ผู้ที่เคยพัฒนาโปรแกรมระบบวินโดว์โดยใช้ทูลคิทที่มีใน Development Tools Kit แล้วพบว่า การพัฒนาต่าง ๆ จะทำได้ง่ายกว่าวิธีการเขียนโปรแกรมประยุกต์ในแบบที่เคยทำมาแล้ว ทั้งนี้เพราะเครื่องมือและโมดูลต่าง ๆ ผู้เขียนโปรแกรมเรียกใช้ได้เลย การเรียกใช้นี้จึงทำให้ผลงานที่ออกมามีขอบเขตการ

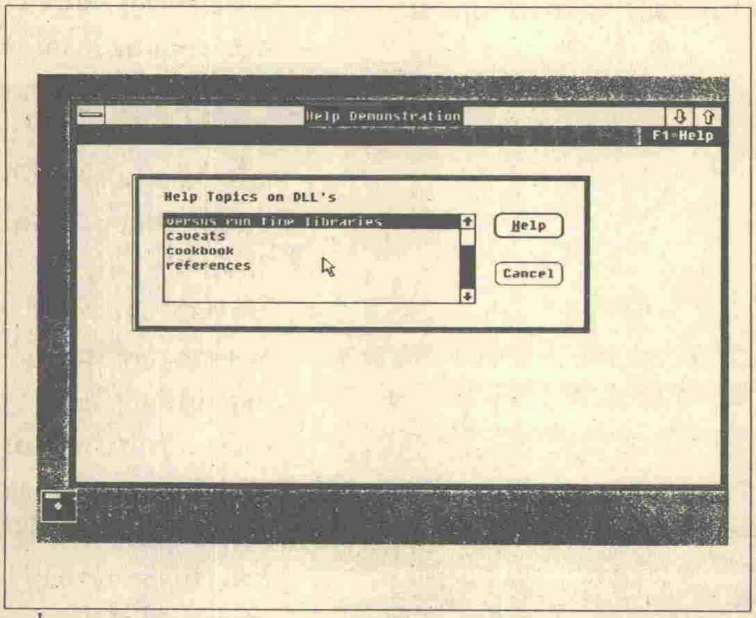


รูปที่ 5 ไมโครซอฟต์วินโดว์ 386 ทำให้การเรียกโปรแกรมประยุกต์ได้หลายโปรแกรมในหลายวินโดว์ การเลือกเมนูใช้เมาส์เป็นหลัก

ใช้งานเสมือนผู้เขียนโปรแกรมมีประสบการณ์มาแล้วมากมาย รูปแบบของการแสดงผลเป็นไปตามมาตรฐานต่าง ๆ

สิ่งที่น่าจับตาคือระบบเอกซ์

เอกซ์วินโดว์เป็นระบบ GUI อีกกลุ่มหนึ่งซึ่งพัฒนาโดยกลุ่มนักวิจัยแห่งมหาวิทยาลัยเอ็มไอที พัฒนาบนสิ่งแวดล้อมโอเอสยูนิกซ์ ปัจจุบันได้พัฒนามาถึงเวอร์ชัน XII เป็นระบบ GUI ที่แพร่หลายอย่างรวดเร็วบนเครื่องเวอร์กสเตชันที่ใช้ยูนิกซ์ โดยจุดสำคัญที่ทำให้แพร่หลายคือเป็นโปรแกรม GUI ที่ค่อนข้างจะไม่ขึ้นกับฮาร์ดแวร์ระบบเครื่องใด เป็นตัวที่โอนย้ายโปรแกรมประยุกต์ไปใช้ได้โดยง่าย



รูปที่ 6 การใช้ DLL ในระบบขอความช่วยเหลือแบบออนไลน์ ทำให้โปรแกรมประยุกต์หลายโปรแกรมเรียก help kernel ตัวเดียวกันได้

รูปที่ 7 ระบบเอกซ์วินโดว์เป็นระบบที่กำลังมาแรงปัจจุบันเป็นระบบ GUI ที่ใช้บนเวอร์กสเตชัน

จุดเด่นของเอกซ์วินโดว์คือ ได้รับการออกแบบมาบนสิ่งแวดล้อมที่เป็นเน็ตเวอร์ก ลักษณะการใช้งานในยุคต่อไปของเวอร์กสเตชันจะเป็นเน็ตเวอร์กมากขึ้น และที่สำคัญคือ เครื่องคอมพิวเตอร์ต่างยี่ห้อกัน เช่น พีซีและเวอร์กสเตชันต่าง ๆ เมื่ออยู่บนเน็ตเวอร์กเดียวกันและอยู่ภายใต้เอกซ์วินโดว์ จะทำให้การทำงานร่วมกันได้อย่างมีประสิทธิภาพ รายละเอียดเกี่ยวกับเอกซ์วินโดว์นั้นหาอ่านได้จากหนังสือ X-Window ที่มหาวิทยาลัย MIT จัดพิมพ์

พัฒนาการในเมืองไทย

ในเมืองไทยยังมีการพัฒนาโปรแกรมประยุกต์ภายใต้สิ่งแวดล้อมของ GUI กันน้อยมาก อย่างไรก็ตามแนวโน้มการพัฒนาในด้านนี้จะหลีกเลี่ยงอีกไม่ได้แล้ว การพัฒนาระบบภาษาไทยภายใต้สิ่งแวดล้อม GUI จะเป็นทางออกที่ดีทางหนึ่ง เพราะการแสดงผลภาษาไทยที่ไม่ใช่ฮาร์ดแวร์ แต่ใช้รูปแบบตัวอักษรกราฟิก ทำให้ระบบการแสดงผลเป็นไปได้อย่างดี และเป็นธรรมชาติตามโครงสร้างภาษาไทยได้ ด้วยเหตุนี้จึงน่าจะมีการส่งเสริมให้ศึกษาและพัฒนาซอฟต์แวร์ภายใต้ GUI ซึ่งไม่ว่าจะเป็นระบบใดก็ได้ หนทางของการพัฒนานั้นยังยาวนานอีกมาก หากพวกเราร่วมกันเราก็จะได้ซอฟต์แวร์ประยุกต์ที่มีคุณภาพต่อไปได้